

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**Отчет по лабораторной работе №2.18
Работа с переменными окружения в Python3
по дисциплине «Технологии программирования»**

Выполнил студент группы ИВТ-б-о-20-1

Ищенко Т.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверила Воронкин Р.А. _____

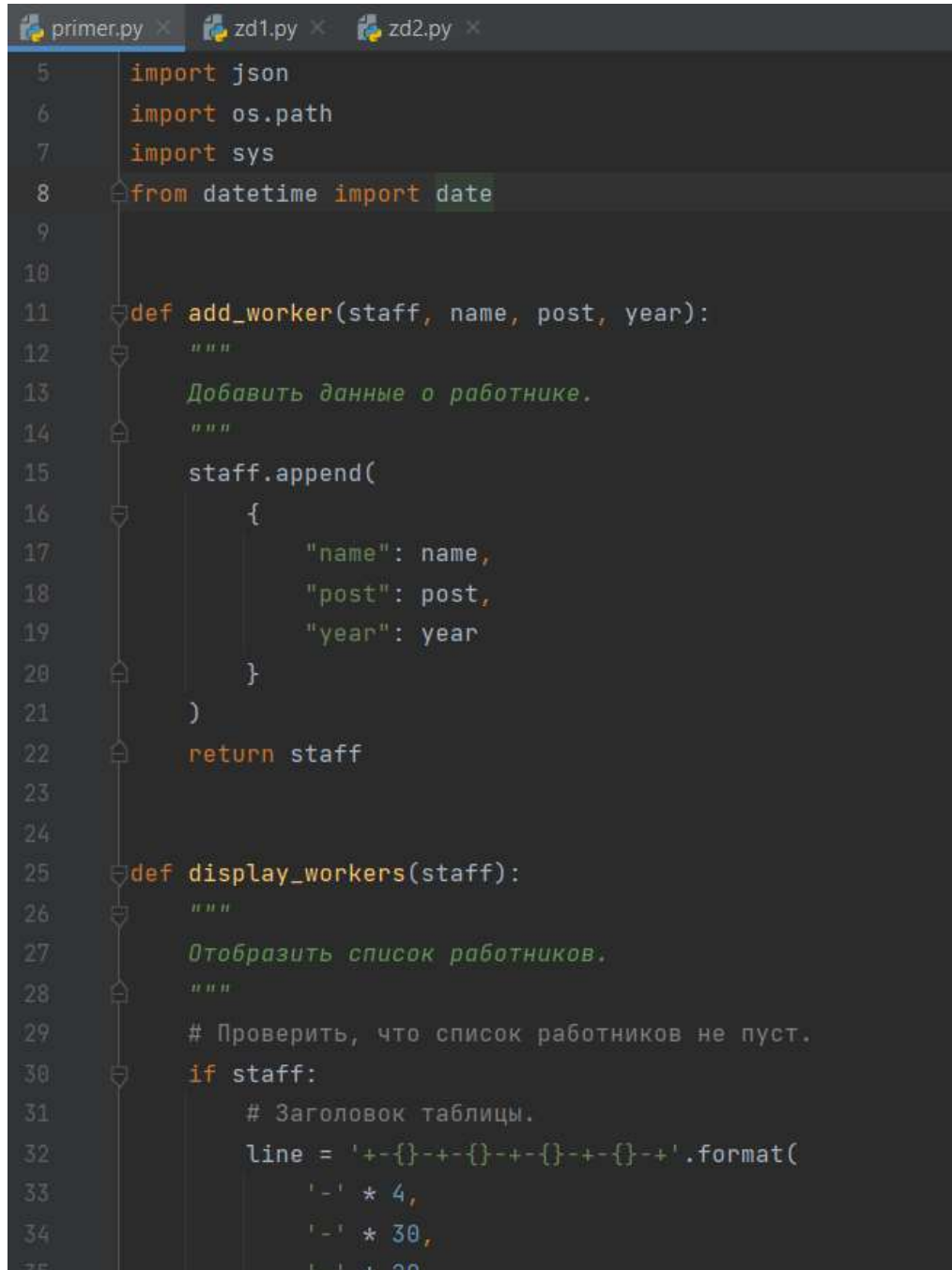
(подпись)

Ставрополь 2021

Цель работы: приобретение навыков по работе с переменными окружения с помощью языка программирования Python версии 3.x.

Ход работы:

1. Произвёл отработку примера, согласно методическим рекомендациям



```
5 import json
6 import os.path
7 import sys
8 from datetime import date
9
10
11 def add_worker(staff, name, post, year):
12     """
13     Добавить данные о работнике.
14     """
15     staff.append(
16         {
17             "name": name,
18             "post": post,
19             "year": year
20         }
21     )
22     return staff
23
24
25 def display_workers(staff):
26     """
27     Отобразить список работников.
28     """
29     # Проверить, что список работников не пуст.
30     if staff:
31         # Заголовок таблицы.
32         line = '+-{}-+-{}-+-{}-+-{}-+'.format(
33             '-' * 4,
34             '-' * 30,
35             '-' * 20,
```

Рисунок 1 – Результат выполнения примера

2. Произвёл выполнение индивидуального задания

```

17
18     = subparsers.add_parser(
19         'display',
20         description=[file_parser],
21         help="Display all stations"
22     )
23
24     select = subparsers.add_parser(
25         'select',
26         description=[file_parser],
27         help="Select the stations"
28     )
29
30     select.add_argument(
31         "-n",
32         "--name_station",
33         action="store",
34         type=int,
35         required=True,

```

```

PS C:\lay\work\scfy\aim_3\tach_prog\2.18\files> python zd1.py add -p="Stavropol" -n="15" -t="10"
PS C:\lay\work\scfy\aim_3\tach_prog\2.18\files> python zd1.py display

```

No	Город прибытия	Номер поезда	Время отправления
1	Stavropol	15	10

```

PS C:\lay\work\scfy\aim_3\tach_prog\2.18\files>

```

Рисунок 2 – Результат выполнения индивидуального задания

3. Выполнил задание повышенной сложности, согласно методическим рекомендациям

```

12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

PS C:\lay\work\scfy\aim_3\tach_prog\2.18\files> python zd2.py add -p="Stavropol" -n="2" -t="12"
PS C:\lay\work\scfy\aim_3\tach_prog\2.18\files> python zd2.py display

```

No	Город прибытия	Номер поезда	Время отправления
2	Stavropol	20	12
3	Stavropol	22	12

Рисунок 3 – Результат выполнения задания повышенной сложности

4. Содержимое файла .env

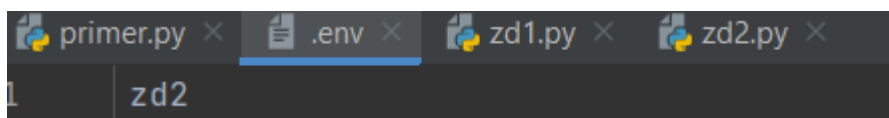


Рисунок 4 – Содержимое файла .env

Контрольные вопросы:

1. Каково назначение переменных окружения? – Переменная среды (переменная окружения) – это короткая ссылка на какой-либо объект в системе. С помощью таких сокращений, например, можно создавать универсальные пути для приложений, которые будут работать на любых ПК, независимо от имен пользователей и других параметров.

2. Какая информация может храниться в переменных окружения? – Переменные среды хранят информацию о среде операционной системы. Эта информация включает такие сведения, как путь к операционной системе, количество процессоров, используемых операционной системой, и расположение временных папок.

3. Как получить доступ к переменным окружения в ОС Windows?

Получить информацию о существующих переменных можно в свойствах системы. Для этого кликаем по ярлыку компьютер и переходим в пункт свойства системы. Далее дополнительные параметры системы и нажимаем на кнопку переменные среды.

4. Каково назначение переменных PATH и PATHEXT? – «PATH» позволяет запускать исполняемые файлы и скрипты, «лежащие» в определенных каталогах, без указания их точного местоположения. PATHEXT, в свою очередь, дает возможность не указывать даже расширение файла, если оно прописано в ее значениях.

5. Как создать или изменить переменную окружения в Windows? – Чтобы добавить или изменить переменные среды, пользователь выбирает пункт система на панели управления, а затем выбирает вкладку Среда. Пользователь также может добавлять или изменять переменные среды в командной строке с помощью команды Set. Переменные среды, созданные с помощью команды Set, применяются только к командному окну, в котором они заданы, и к его дочерним процессам.

6. Что представляют собой переменные окружения в ОС Linux? – Переменные окружения в Linux представляют собой набор именованных значений, используемых другими приложениями. Переменные окружения применяются для настройки поведения приложений и работы самой системы. Например, переменная окружения может хранить информацию о путях к исполняемым файлам, заданном по умолчанию текстовом редакторе, браузере, языковых параметрах системы или настройках раскладки клавиатуры.

7. В чем отличие переменных окружения от переменных оболочки? – Переменные окружения (или «переменные среды») — это переменные, доступные в масштабах всей системы и наследуемые всеми дочерними процессами и оболочками. Переменные оболочки — это переменные, которые применяются только к текущему экземпляру оболочки. Каждая оболочка, например, `bash` или `zsh`, имеет свой собственный набор внутренних переменных.

8. Как вывести значение переменной окружения в Linux? – Значение каждой переменной окружения изначально представляет собой строковую константу (строку). Интерпретация значений переменных полностью возлагается на программу. Иными словами, все переменные окружения имеют тип `char*`, а само окружение имеет тип `char**`. Чтобы вывести на экран значение какой-нибудь переменной окружения, достаточно набрать `echo $ИМЯ_ПЕРЕМЕННОЙ`.

9. Какие переменные окружения Linux Вам известны?

`USER` — текущий пользователь.

`PWD` — текущая директория.

`OLDPWD` — предыдущая рабочая директория. Используется оболочкой для того, чтобы вернуться в предыдущий каталог при выполнении команды

`cd`.

HOME — домашняя директория текущего пользователя.

SHELL — путь к оболочке текущего пользователя (например, bash или zsh).

EDITOR — заданный по умолчанию редактор. Этот редактор будет вызываться в ответ на команду edit.

LOGNAME — имя пользователя, используемое для входа в систему.

PATH — пути к каталогам, в которых будет производиться поиск вызываемых команд. При выполнении команды система будет проходить по данным каталогам в указанном порядке и выберет первый из них, в котором будет находиться исполняемый файл искомой команды.

LANG — текущие настройки языка и кодировки.

TERM — тип текущего эмулятора терминала.

MAIL — место хранения почты текущего пользователя.

LS_COLORS — задает цвета, используемые для выделения объектов — (например, различные типы файлов в выводе команды ls будут выделены разными цветами).

10. Какие переменные оболочки Linux Вам известны?

BASHOPTS — список задействованных параметров оболочки, разделенных двоеточием.

BASH_VERSION — версия запущенной оболочки bash.

COLUMNS — количество столбцов, которые используются для отображения выходных данных.

DIRSTACK — стек директорий, к которому можно применять команды pushd и popd.

HISTFILESIZE — максимальное количество строк для файла истории команд.

HISTSIZE — количество строк из файла истории команд, которые можно хранить в памяти.

HOSTNAME — имя текущего хоста.

IFS — внутренний разделитель поля в командной строке (по умолчанию используется пробел).

PS1 — определяет внешний вид строки приглашения ввода новых команд.

PS2 — вторичная строка приглашения.

SHELLOPTS — параметры оболочки, которые можно устанавливать с помощью команды `set`.

UID — идентификатор текущего пользователя.

11. Как установить переменные оболочки в Linux? – Команда `set` — устанавливает переменные оболочки. При использовании без аргумента выведет список всех переменных, включая переменные окружения и переменные оболочки, а также функции оболочки.

12. Как установить переменные окружения в Linux? – Команда `export` используется для задания переменных окружения. С помощью данной команды мы экспортируем указанную переменную, в результате чего она будет видна во всех вновь запускаемых дочерних командных оболочках. Переменные такого типа принято называть внешними.

13. Для чего необходимо делать переменные окружения Linux постоянными? – Если вы хотите, чтобы переменная сохранялась после закрытия сеанса оболочки, то необходимо прописать её в специальном файле. Прописать переменную можно как для текущего пользователя, так и для всех пользователей.

14. Для чего используется переменная окружения `PYTHONHOME`? – Переменная среды `PYTHONHOME` изменяет расположение стандартных библиотек Python. По умолчанию библиотеки ищутся в `prefix/lib/pythonversion` и `exec_prefix/lib/pythonversion`, где `prefix` и `exec_prefix` - это каталоги, зависящие от установки, оба каталога по умолчанию - `/usr/local`.

15. Для чего используется переменная окружения `PYTHONPATH`? – Переменная среды `PYTHONPATH` изменяет путь поиска по умолчанию для

файлов модуля. Формат такой же, как для оболочки PATH: один или несколько путей к каталогам, разделенных `os.pathsep` (например, двоеточие в Unix или точка с запятой в Windows). Несуществующие каталоги игнорируются.

16. Какие еще переменные окружения используются для управления работой интерпретатора Python?

PYTHONSTARTUP: Если переменная среды **PYTHONSTARTUP** это имя файла, то команды Python в этом файле выполняются до отображения первого приглашения в интерактивном режиме. Файл выполняется в том же пространстве имен, в котором выполняются интерактивные команды, так что определенные или импортированные в нем объекты можно использовать без квалификации в интерактивном сеансе. При запуске вызывает событие аудита `cpython.run_startup` с именем файла в качестве аргумента.

PYTHONOPTIMIZE: Если в переменной среды **PYTHONOPTIMIZE** задана непустая строка, это эквивалентно указанию параметра `-O`. Если установлено целое число, то это эквивалентно указанию `-OO`.

PYTHONBREAKPOINT: Если переменная среды **PYTHONBREAKPOINT** установлена, то она определяет вызываемый объект с помощью точечной нотации. Модуль, содержащий вызываемый объект, будет импортирован, а затем вызываемый объект будет запущен реализацией по умолчанию `sys.breakpointhook()`, которая сама вызывается встроенной функцией `breakpoint()`. Если **PYTHONBREAKPOINT** не задан или установлен в пустую строку, то это эквивалентно значению `pdb.set_trace`. Установка этого значения в строку `0` приводит к тому, что стандартная реализация `sys.breakpointhook()` ничего не делает, кроме немедленного возврата.

17. Как осуществляется чтение переменных окружения в программах на языке программирования Python? – Значения переменных окружения можно считывать и изменять при помощи объекта `environ` модуля `os` либо путем использования функций `setdefault()` и `get()`. В качестве ключа, по которому

можно обратиться и получить либо присвоить значение переменной, в `environ` используется имя переменной окружения.

18. Как проверить, установлено или нет значение переменной окружения в программах на языке программирования Python? – С помощью следующего кода:

```
if os.environ[key_value]:  
    print(  
        "The value of",  
        key_value,  
        " is ",  
        os.environ[key_value]  
    )
```

19. Как присвоить значение переменной окружения в программах на языке программирования Python? – С помощью следующего кода:
`os.environ.setdefault('DEBUG', 'True')`

Вывод: в ходе выполнения лабораторной работы приобрел навыки по работе с переменными окружения с помощью языка программирования Python версии 3.x.